

# (ALU) گزارش پروژه: طراحی واحد محاسبات و منطق

## گزارش پروژه: طراحی واحد محاسبات و منطق (ALU)

### مقدمه

در این پروژه، یک واحد محاسبات و منطق (ALU) طراحی شده است که عملیات منطقی و ریاضی را روی دو عدد ۸ بیتی اجرا می‌کند. ALU یکی از اجزای اصلی پردازنده‌ها است و در انجام محاسبات مختلف نقش کلیدی دارد. این ماژول قابلیت انجام ۱۶ عملیات متفاوت را دارد که شامل پردازش داده‌های منطقی و حسابی است. همچنین، طراحی ماژول به صورت مدولار انجام شده تا خوانایی و قابلیت توسعه افزایش یابد.

## ساختار واحد ALU

ماژول ALU شامل سه بخش اصلی است:

- را انجام می‌دهد AND، OR، XOR، NAND عملیات منطقی مانند - واحد منطقی (Logic Unit)
- عملیات محاسباتی مانند جمع، تفریق، افزایش و کاهش را مدیریت می‌کند - واحد حسابی (Arithmetic Unit)
- تصمیم می‌گیرد که خروجی مربوط به واحد منطقی یا واحد حسابی را (Opcode) واحد مدیریت و انتخاب خروجی - بر اساس کد عملیات ارسال کند.

## توضیح روش تبدیل عملیات

ماژول ALU از یک سیگنال ۴ بیتی (Opcode) برای انتخاب عملیات استفاده می‌کند. روش پردازش به این صورت است:

عملیات منطقی - بیت‌های دو ورودی a و b با استفاده از عملیات منطقی پردازش می‌شوند:

- a: متمم (0000)
- b: متمم (0001)
- a AND b (0010)
- a OR b (0011)
- a NAND b (0100)
- a NOR b (0101)
- a XOR b (0110)
- a XNOR b (0111)

عملیات حسابی - این بخش شامل دستورات ریاضی روی a و b است:

- a: انتقال (1000)
- b: انتقال (1001)
- به اندازه ۱ افزایش مقدار a (1010)
- به اندازه ۱ افزایش مقدار b (1011)
- به اندازه ۱ کاهش مقدار a (1100)
- به اندازه ۱ کاهش مقدار b (1101)

- **a** و **b** جمع: **(1110)**
- **(cin)** با حمل **b** و **a** جمع: **(1111)**

## مدولار بودن طراحی

در طراحی این ALU از رویکرد **مدولار** استفاده شده است. به این معنا که:

- **واحد منطقی و واحد حسابی** در فایل‌های جداگانه پیاده‌سازی شده‌اند.
- انتخاب بین این دو واحد از طریق **واحد مدیریت خروجی** انجام می‌شود.
- این روش **خوانایی** کد را بهبود می‌بخشد و امکان **افزودن عملیات جدید** را بدون تغییر در کل سیستم فراهم می‌کند.

## تست و اعتبارسنجی

برای بررسی صحت عملکرد ALU، یک **Test Bench** طراحی شد. این تست پنج تمامی کدهای عملیات (**Opcode**) را بررسی کرده و خروجی ALU را در شرایط مختلف نمایش می‌دهد.

- مقدار اولیه متغیرها:

**a = 00001111**

**b = 00010001**

**cin = 0**

- مجموعه تست‌ها:

1. بررسی تمام عملیات منطقی.
2. بررسی عملیات انتقال، افزایش و کاهش مقدار.
3. تست عملیات جمع ساده و جمع همراه با حمل.
4. به تغییرات ورودی‌ها **ALU** بررسی واکنش.

## نتیجه‌گیری

ماژول ALU طراحی شده، عملکرد مورد انتظار را به‌طور **کامل و صحیح** ارائه می‌دهد. با استفاده از **مدل طراحی مدولار**، امکان **گسترش و بهینه‌سازی** در آینده فراهم شده است. این ALU می‌تواند در سیستم‌های پردازشی و محاسباتی به عنوان یک **واحد عملیاتی اصلی** مورد استفاده قرار گیرد.

مزایای این طراحی عبارت‌اند از:

- ✓ **مدولار بودن** - تفکیک وظایف و افزایش خوانایی کد
- ✓ **تست‌پذیری بالا** - تست خودکار تمامی عملیات
- ✓ **امکان توسعه** - قابلیت افزودن عملیات جدید
- ✓ **بهینه بودن** - استفاده صحیح از منابع سخت‌افزاری